Reliable Recommendation with Review-level Explanations

Yanzhang Lyu[§] Hongzhi Yin[†¶] Jun Liu[§] Mengyue Liu[§] Huan Liu[§] Shizhuo Deng[‡]

[§]*National Engineering Lab for Big Data Analytics, Xi'an Jiaotong University*
School of Electronic and Information Engineering, Xi'an Jiaotong University
[†]*School of Information Technology and Electrical Engineering, The University of Queensland*
[‡]*School of Computer Science and Engineering, Northeastern University*

[§]yanzhanglyu@163.com liukeen@xjtu.edu.cn
liumengyue@stu.xjtu.edu.cn huanliucs@gmail.com
[†]h.yin1@uq.edu.au
[‡]dengshizhuo@gmail.com

***Abstract*—The quality of user-generated reviews is significant for users to understand recommendation results and make online purchasing decisions correctly. However, the reliability of a review, which captures the likelihood that a review is benign, is ignored by many studies. The low reliability reviews cause a recommendation system's unsatisfying performance. Especially the fake reviews written by fraudulent users mislead the system into generating error recommendation results and explanations, which confuse customers and deprive customers of confidence in the system. In this paper, we propose a model, Reliable Recommendation with Review-level Explanations (RRRE), which detects reliable reviews and improves the performance of the explainable recommendation system as well. Recognizing the textual content of reviews, user-item interactions are valuable features for both rating prediction and reliability prediction. RRRE builds a uniform framework to predict rating scores and reliability scores simultaneously. Firstly, RRRE embeds user preferences and item profiles, which are extracted from textual and interactive features, into the representation of the review. Secondly, the supervised information of two subtasks is jointly combined. It makes the optimization of RRRE faster and better. Finally, the reviews with both high reliability scores and rating scores are given to customers as reliable explanations. To the best of our knowledge, we are the first to consider the reliability of reviews for improving explainable recommender system. And the experimental results confirm this idea and show that our model outperforms other baseline methods on Yelp and Amazon datasets.**

***Index Terms*—recommender system, explainable recommendation, rating prediction, attention network, review-reliability**

I. Introduction

With the development of e-commerce, people have become accustomed to picking products on-line rather than in physical stores. Online shopping brings users more choices but also information overload. Recommender systems have been widely adopted by many websites to help users making decisions. Many systems based on the Collaborative Filtering (CF) technique and Neural Network have shown good performance in generating recommended results, but they can not explain why an item is recommended [1]–[4]. Without the support of evidence, recommendation results are easily questioned by customers. Therefore, explainable recommender systems are proposed to enhance customers' satisfaction and sense of trust by explaining how the system works [5], [6].

Many recent studies explore user reviews for mining user preferences and generating a proper explanation. For example, EFM [7] and LRPPM [8] extract explicit product features and user opinions by phrase-level sentiment analysis on user reviews, then generate both recommendation results and phrase-level explanations. NARRE [9] detects the most valuable reviews to improve the performance of the recommender system and explains the rating prediction process. DER [10] designs a time-aware and sentence-level neural network to model users' dynamic preferences.

However, user-generated reviews are useful but not completely reliable [11]–[13]. In real scenarios, many users give fake reviews because of a huge monetary incentive. In this paper, the **reliability** of a review is defined as the likelihood that a review is benign. The reliability score is a continuous value from 0 to 1, '1' means a review is benign, and '0' means it is fake. Low reliability reviews are written to unjustly promote bad items and demote good items. Low reliability in reviews is not an isolated phenomenon and is usually intertwined with biased ratings [14] [15]. Thus, the review with low reliability makes less contribution to the item profiles and the user preferences for the recommendation. Although the former explainable recommender models have achieved good results, none of them considered the reliability of reviews. Such negligence brings two limitations. First, fake reviews reduce the performance of recommender systems. The more accurately the system learns from fake reviews, the more incorrect the recommendation results will be. Second, fake reviews reduce the explaining ability of an explainable recommender system. When customers receive a fake review as an explanation, they will feel cheated and question the efficiency and the motivation of the recommender system.

[¶]Corresponding author; contributing equally with the first author.

To learn the reliability of the reviews and improve the performance of the explainable recommender system, we propose a model named Reliable Recommendation model with Review-level Explanations (RRRE), which can calculate rating score for recommendation and reliability score for explanation in a uniform model. This model is inspired by a widely used assumption in fraud detection [16]: An item is good (bad) if most of the reviews it receives are positive (negative) benign-reviews or negative (positive) fake-reviews, and a user is benign if he mostly writes positive (negative) reviews to good (bad) items. This paper extends the assumptions with the definition of **reliability**: A review has high reliability when the review is written from a benign user towards a good item. It is obvious that the reliability of a review is decided by its relevant user preferences and item profiles, which are also important for the recommendation task. Therefore, a parallel neural network model is designed to learn a pair of user embedding and item embedding as the representation for a review. An embedding layer extracts the user preferences and the item profiles from the textual content of the user's (item's) reviews. An attention mechanism utilizes the interaction between users and items to generate the weighted user (item) embeddings. The labels of benign reviews and ratings are used to optimize the prediction of rating scores and reliability scores.

The contributions and advantages of this paper can be summarized as follows:

- We propose a neural network-based framework, Reliable Recommendation with Review-level Explanations, to predict the rating score and the reliability score of reviews simultaneously. RRRE utilizes textual feature and user-item interaction to generate the review embedding, which is effective for both rating score prediction and reliability score prediction. To the best of our knowledge, we are the first to consider the reliability of reviews for improving the performance of recommendation and the ability to explain recommendation results.
- We jointly combine the supervised information of two subtasks. Predicting rating scores of fake reviews is meaningless and may damage the performance of prediction for benign reviews. Therefore, the label of the benign review can promote rating prediction. We also define the biased Root Mean Square Error (bRMSE), which only calculates the RMSE value of benign reviews, as the evaluation metric.
- RRRE employs bRMSE, average precision (AP), AUC, and NDCG@k as evaluation metrics and outperforms state-of-the-art techniques on the real-world review datasets collect from Yelp and Amazon.

In the rest of the paper, we review the related work in Section II, and give a detailed explanation of our methods in Section III. Experiment settings and results are shown in Section IV. In Section V, the conclusion and future work are presented.

II. Related Work

The main task of this paper is to build an explainable recommender system and its related work is surveyed in Section II-A. This paper considers the impact of fake reviews on the recommender system. Therefore, some work about fake review detection is shown in Section II-B. The definition of reliability in this paper is a trust-based metric, so the trust-aware recommender system is also surveyed in Section II-C.

A. Explainable Recommendation

The explainable recommendation is becoming more and more popular in both research and industry communities [17]. The early methods [7], [9] generated a feature-level explanation for recommendation results. These works are mainly based on the combination of phrase-level sentiment analysis and matrix factorization. Their explanations are provided by filling the user cared features into predefined templates. Despite effectiveness, this kind of methods have two natural limitations. Firstly, the performance of these models may be limited by the accuracy of natural language processing (NLP) tools. Secondly, features-level explanations are not easy for people to understand. Besides, predefined explanation templates can't adapt to multiple scenes flexibly.

Many recent studies have focused on the sentence-level explainable recommendation. Wang et al. [18] developed a multi-task learning solution, which integrates two learning tasks, one is the user preferences modeling for recommendation and the other is opinionated content modeling for explanation, via a joint tensor factorization. With the ever prospering of deep learning technology, many algorithms based on the attention mechanism [9], [10] were designed. NARRE [9] considers the usefulness of reviews for recommendation quality and proposes an attention mechanism to explore the usefulness of reviews. DER [10] designs a time-aware gated recurrent unit (GRU) to model users' dynamic preferences. Besides, DER provides explanations tailored for the users' current preferences.

Although these models have achieved promising results, they haven't realized that the fraudulent users and fake reviews in real scenarios would cheat the recommender system badly.

B. Fake Review Detection

The fake review, also known as the fraudulent rating [19] or the opinion spam [20], often unjustly promotes or demotes certain products. The reliability of a review describes the likelihood that a review is benign. So the fake review detection is reviewed here as the related work of reliability score prediction.

Existing work in fake review detection can be categorized into network-based, content-based, and behavior-based:

1) Network-based Methods: The networks in this category of work are graphs consisting of users, items, ratings, and reviews. Li et al. [21] defined a vector-valued contractive function to characterize the trustworthiness and the importance of a node in trust networks. Wang et al. [22] explored how interactions between nodes in the review graph could reveal the

cause of spam and proposed an iterative computation model to identify suspicious reviewers. Wu et al. [23] proposed a node ranking technique that models the probability of the trustworthiness of individual data sources as an interpretation for the underlying ranking values. Jiang et al. [24] identified group spammers solely based on their abnormal network footprints.

2) Content-based Methods: This category focuses on learning how the characteristics of language that fake reviews use differ from benign reviews. Fayazi et al. [25] proposed a three-part method to uncover the crowdsourced manipulation of online reviews. Sandulescu et al. [26] calculated the semantic similarity to find similar reviews for singleton review spammer detection. Sun et al. [27] leveraged the difference of semantic flows between synthetic and truthful reviews, then developed a defense method to improve the performance of fake review detection. A neural network model was explored by Ren et at. [28] to learn document-level representation for detecting deceptive opinion spam.

3) Metadata-based Methods: The methods in this category often utilize indicative features extracted from the metadata associated with user preferences, item profiles, statistical information of review, timestamp, and location. Mukherjee et al. [29] proposed an unsupervised model to capture the behavioral distributions of opinion spammers. Xie et al. [30] observed the relationship between the fake review and its corresponding rating, and discovered the temporal pattern of fake reviews. SpEagle [20] and FraudEagle [16] incorporated all metadata (text, timestamp, rating) as well as network information under a unified framework to spot suspicious reviews.

In this paper, RRRE extracts semantic information from textual content and calculates the fraud-attention to generate user embeddings and item embeddings, which are used to jointly predict the rating scores and the reliability scores.

C. Trust-aware Recommender System

In the trust-aware recommender system (TARS), there exists two major types of approaches: memory-based and model-based [31] methods, which are described below.

1) Memory-based Methods: This kind of methods use the trust relation between two users instead of similarity to generate recommendation results. Massa et al. [32] proposed a trust inference method in a binary trust network only considering the shortest distance between two users and maximum trust propagation length. TrustWalker [33] is a random walk model that combines the item-based ranking with the trust-based nearest neighbor model.

2) Model-based Methods: This kind of approaches in TARS use matrix factorization techniques to incorporate trust information. Ma et al. [34] linearly combined matrix factorization of rating matrix with trusted neighbors, along with the strategy of sharing common user-feature matrix. Yang et al. [35] adopted matrix factorization to map users into low-dimensional latent feature spaces in terms of their trust relationship. TrustSVD [36] integrates multiple information sources into the recommendation model to reduce the data sparsity and cold-start problems.

Both the trust-aware methods and RRRE aim at the reliable recommendation. This paper focuses on the reliability of reviews rather than the trust relation between users. In addition, these trust-aware recommendation methods are constrained by their limited explainability.

TABLE I
NOTATIONS

Symbols	Descriptions
t^{ui}	Tuple of a review from u to i, $t^{ui} = \{u, i, r_{ui}, l_{ui}, w_{ui}\}$
r_{ui}	Rating score of t^{ui}
l_{ui}	Reliability score of t^{ui}
w_{ui}	Textual content of t^{ui}
rev_{ui}	The embedding of review w_{ui}
W^i	The set of review contents that written to a item i
W^u	The set of review contents that written by an user u
$UserNet$	The network which generate the embeddings of users
$ItemNet$	The network which generate the embeddings of items
x_u	The embedding of a user u
y_i	The embedding of an item i
k	hyper-parameter: the dimension of rev^w
m	hyper-parameter: the number of reviews in input layer

III. METHODOLOGY

This section formulates our problem and describes the details of Reliable Recommendation with Review-level Explanations (short for RRRE). Firstly, the framework of RRRE is introduced. The next several subsections elaborate the major components and technical details of our model. At last, we will go through the optimization details of RRRE.

A. Problem Definition

Suppose there is a set of users $U = \{u_1, u_2, ..., u_n\}$ and a set of items $I = \{i_1, i_2, ..., i_m\}$. A tuple $t^{ui} = \{u, i, r_{ui}, l_{ui}, w_{ui}\}$ means a review from a user u to an item i with a rating r_{ui} and a reliability score l_{ui}, w_{ui} means the textual content of the review. Given all the information of reviews in the training set $T = \{t^{ui} | u \in U, i \in I\}$, our task is to find a mapping function f that satisfies the following equation

$$f(u, i, W^u, W^i) = (r_{ui}, l_{ui}) \tag{1}$$

where $W^u = \{w^{ui} | i \in I\}$ and $W^i = \{w^{ui} | u \in U\}$.

The mathematical notations used in this paper are summarized in TABLE I.

B. Framework

The framework of the proposed model for rating prediction is shown in Fig. 1. RRRE consists of three main components, review embedding, user & item embedding, and score prediction. The first two components have two parallel networks $UserNet$ and $ItemNet$ corresponding to users and items respectively. $UserNet$ and $ItemNet$ have the same workflow, the former is to learn the user preferences and the latter to learn the item profiles. The two parallel networks utilize W^u of a user u and W^i of an item i as inputs in the first component, and

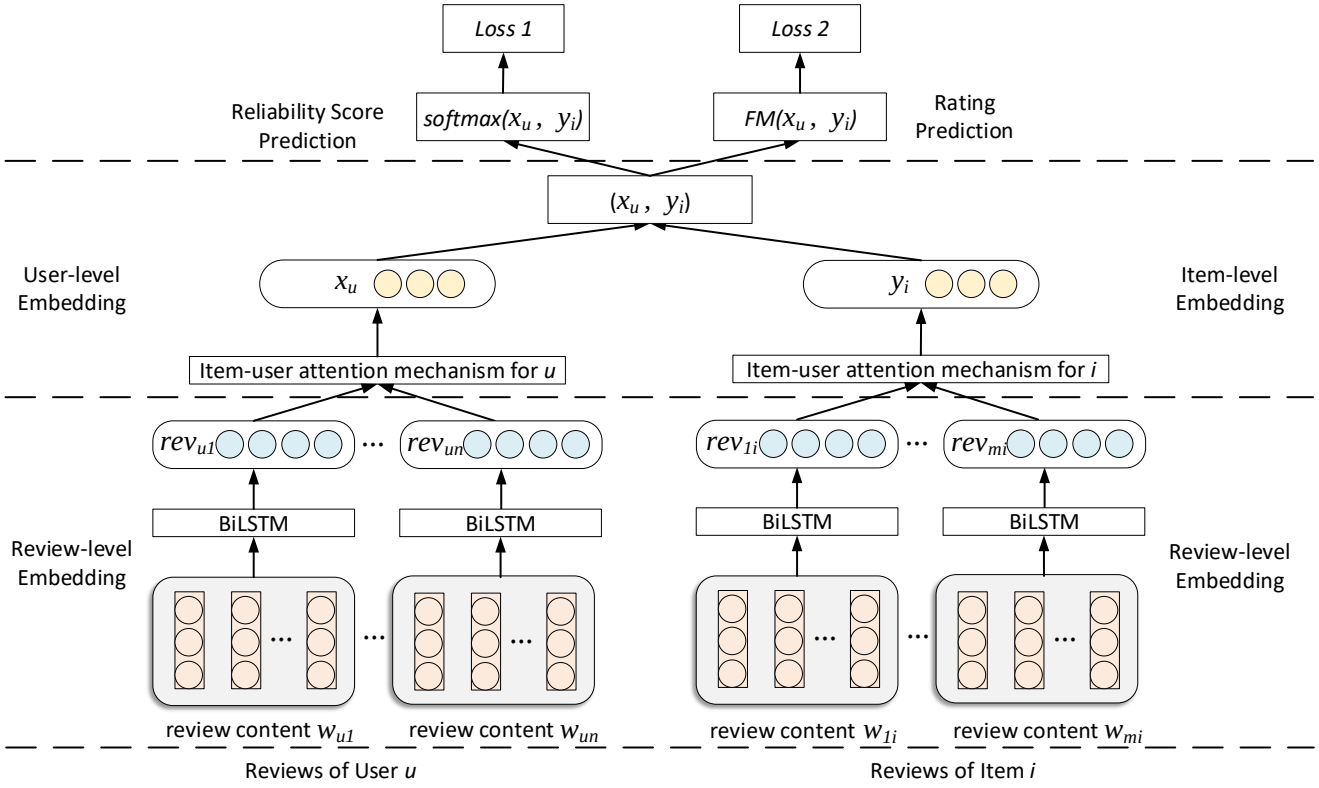

Fig. 1. The architecture of Reliable Recommendation with Review-level Explanations neural network

embed each textual content w_{ui} as rev_{ui}. Then the next layer weights different reviews by a fraud-attention mechanism and generates an embedded user-item pair (x_u, y_i). In the third component, x_u and y_i are jointly combined to predict the rating score and the reliability score. Two loss functions are designed to optimize the prediction of rating score r_{ui} and reliability score l_{ui} as final results.

The generating procedure of the recommendation results and the explanations is based on the predicted rating score r_{ui} and reliability score l_{ui}. RRRE calculates $R^{u_0} = \{(r_{u_0 i}, l_{u_0 i}) | i \in I\}$ for each user u_0, then sorts the R^{u_0} by $r_{u_0 i}$ and takes the top $\mathcal{K}$ as the candidate set $R^{u_0}_{\mathcal{K}}$. Finally, RRRE sorts $R^{u_0}_{\mathcal{K}}$ by $l_{u_0 i}$ and takes the top $\mathcal{K}$ elements as the recommendation results for u_0. For each recommended item i_0, we calculate $R^{i_0} = \{(r_{ui_0}, l_{ui_0}) | u \in U\}$, then sort R^{i_0} according to the above strategy and take the textual content of the top $\mathcal{K}$ elements as the explanations of i_0.

C. Review Content Embedding

In recent years, many word embedding approaches have boosted the performance in many NLP applications [37], [38]. To capture semantic information from reviews, RRRE processes text using a word embedding function h to map each word into a d dimensional vector. Each review is denoted as a document $d^w_\theta = \{word^w_1, word^w_2, ..., word^w_\theta\}$. Then, a matrix of long word embedding sequences V^w_θ is built for review w_{ui} as follows:

$$v^w_\theta = h(word^w_\theta) \tag{2}$$

$$V^w_\theta = v^w_1 \oplus v^w_2 \oplus ... \oplus v^w_\theta \tag{3}$$

where $\oplus$ is the concatenation operator to preserve the sequence information of words.

RRRE employs LSTM, which is widely used for text modeling because of its excellent performance on modeling long documents, to address the problem of long-term dependency. The matrix V^w_θ is put in LSTM unit as a sequence $\{v^w_1, v^w_2, ..., v^w_\theta\}$. In order to increase the amount of input information available to LSTM, a more common approach is to adopt bidirectional LSTM (BiLSTM) to model text semantics both from forward and backward. The forward $LSTM^+$ reads sequence from v^w_1 to v^w_θ and the backward $LSTM^-$ reads sequence from v^w_θ to v^w_1. Thus, the context information is summarize in $rev^+_{ui} = LSTM^+(V^w_\theta)$ and $rev^-_{ui} = LSTM^-(V^w_\theta)$. The final embedding of w_{ui} comes from a concatenation operator:

$$rev_{ui} = rev^+_{ui} \oplus rev^-_{ui} \tag{4}$$

D. User & Item Embedding

In real scenarios, the different reviews written by the same user contribute to the user preferences with different importances [39]. In order to address this issue, RRRE uses attention

mechanism to distinguish the content which is significant to the properties of users. The final representation for u or i is aggregated by the weighted textual content. In the first layer, the input contains the feature vector rev_{uj} of the jth review of a user u, the item that it written for (ID embedding, $\mathbf{e}^i$), and the user who writes it (ID embedding, $\mathbf{e}^u$). The ID embeddings are added to identify the reliability of users and items simultaneously. Specifically, the attention weight α_{uj} for each review in $UserNet$ can be defined as:

$$\alpha^*_{uj} = h^T tanh(W_{rev} rev_{uj} + W_u \mathbf{e}^u + W_i \mathbf{e}^i + b_1) + b_2, \quad (5)$$

$$\alpha_{uj} = \frac{exp(\alpha^*_{uj})}{\sum_{j=1} exp(\alpha^*_{uj})}, \quad (6)$$

where h is weight vector and h^T represents its transpose. W_{rev}, W_u, W_i are weight matrices, α^*_{uj} is a score function which scores the reliability of jth reviews of a user u.

After obtaining the attention weight of each review, the weighted representation vector u_n for a user u is calculated by weighted sum function. Then the final representation x_u is computed by a fully connected layer with weight matrix W_f and bias b_f:

$$u_n = \sum_{j=1}^{s} \alpha_{uj} rev_{uj}, \quad (7)$$

$$x_u = W_f u_n + b_f. \quad (8)$$

Similarly, RRRE captures the item profiles from weighted textual content written to the item. Therefore, the item i's representation y_i is calculated through $ItemNet$ with the attention mechanism described above.

Practically, the numbers of users' (items') reviews are variable among different datasets. But the scale of $UserNet$ ($ItemNet$) should be determined before the training process. In order to solve this problem, a hyper-parameter m is employed to control the number of input data. If $m > |W^u|$ ($m > |W^i|$), RRRE shapes the input data with the zero-padding operation. If $m < |W^u|$ ($m < |W^i|$), RRRE utilizes a time-based sampling strategy to choose m reviews from W^u (W^i). This sampling strategy takes into account that users' preferences for products change over time and the latest preference is more useful. We sort the W^u (W^i) by publishing time and then select the latest m reviews. The setting of m will be discussed in Section IV.

E. Score Prediction

RRRE optimizes the rating score and the reliability score in the uniform framework because of the following reasons. At first, the motivation of this paper is to promote the performance of recommendation and explanation by distinguishing fake and benign reviews. The reliability score is supposed to be computed along with the rating score. Secondly, the user preferences and the item profiles are important for both recommendation and fake review detection. The high-level representations, x_u and y_i, make it possible and proper for RRRE to optimize two targets simultaneously. In order to make full use of x_u and y_i, we concatenate them as the final review representation and draw on a fully-connected layer and a softmax layer to project the user representation x_u into reliability distribution of C classes:

$$p_c(l) = softmax(\mathbf{W}[x_u, y_i] + \mathbf{b}), \quad (9)$$

where $[\cdot, \cdot]$ is the concatenate operation. The reliability score is the likelihood that a review is benign:

$$l_{ui} = p_{benign}(l). \quad (10)$$

Then, the cross-entropy error between ground truth distribution $g_c(l)$ and predicted result $p_c(l)$ is defined as $loss_1$:

$$loss_1 = -\sum_{U \times I} \sum_{c \in C} g_c(l) \cdot log(p_c(l)), \quad (11)$$

where U represents users in the training set and I is the item set. The ground truth $g_c(l)$ is l_{ui} in t^{ui} with value 0 or 1.

As for rating prediction, RRRE further introduces auxiliary embeddings e^u and e^i [9], [10], [40] to derive the final rating from a user u to an item i:

$$\hat{r}_{ui} = FM([(e^u + W_h x_u), (e^i + W_e y_i)]), \quad (12)$$

where W_h, W_e are weighting parameters, and $FM()$ is the factorization machine layer [9], [10]. The predicted score r_{ui} is commonly evaluated by Mean Square Error (MSE), the loss function with a set of regularized parameters is formulated as:

$$lossfunction = \frac{1}{N} \sum_{U \times I} (r_{ui} - \hat{r}_{ui})^2 + \gamma \sum ||\epsilon||_2^2, \quad (13)$$

where N indicates the number of user-item pairs. In this paper, recommender system only provides services for normal users. So predicting rating scores for fake reviews is meaningless. Besides, considering the ratings of fake reviews hurt the model's performance badly. RRRE designs a biased loss function to optimize the rating prediction task:

$$loss_2 = \frac{1}{N} \sum_{U \times I} l_{ui} (r_{ui} - \hat{r}_{ui})^2 + \gamma \sum ||\epsilon||_2^2, \quad (14)$$

where r_{ui} is the ground truth of rating. ϵ is the set of parameters to be regularized. l_{ui} is the ground truth of reliability, this factor is used to prevent the training process from being damaged by fake reviews.

The final loss of our model is a weighted sum of $loss_1$ and $loss_2$:

$$L = \lambda loss_1 + (1 - \lambda) loss_2. \quad (15)$$

IV. EXPERIMENTS

A. Experiment Setup

1) datasets: We use three datasets (YelpChi, YelpNYC, YelpZip) that have been collected and used by [20] to evaluate our model. The datasets are composed of user ID, item ID, the textual content of the review, rating, and label of reliability, which are collected from Yelp [1] with Yelp's filtering algorithm.

[1]Yelp allows users to search for restaurants by zip code, e.g., http://$www.yelp.com/search?cflt$=$restaurants\&find\_loc$=11794

TABLE II
STATISTICS OF THE DATASETS

	#Reviews	#Percentage of Fake Review	#Items	#Users
YelpChi	67,395	13.23%	201	38,063
YelpNYC	359,052	10.27%	923	160,225
YelpZip	608,598	13.22%	5,044	260,277
Musics	70,170	24.93%	24639	16296
CDs	49,085	22.39 %	26290	23572

The Yelp page of a business shows the recommended reviews, while it is also possible to view the filtered/unrecommended reviews through a link at the bottom of the page. We consider them as genuine and fake, respectively. We also use Musics and CDs, which are collected from Amazon [2], for evaluation. The ground truth is based on helpfulness votes. Only the users who receive at least 20 votes are considered in this dataset. A review is assumed to be benign if the fraction of helpful-to-total votes is at least 0.7, and it is assumed to be fraudulent if the fraction is no greater than 0.3. The statistics of the datasets can be seen in Table II. The percentage of the fake reviews in Amazon datasets is more balanced than that in Yelp datasets.

In order to improve the training speed, the textual content of reviews is pretrained as vectors. Because the dimension of feature space has a great influence on the experimental results, RRRE sets a range of vector dimensions and tests the performances in the following experiments.

2) Evaluation Metric: To evaluate the performance of all algorithms for rating prediction, we extend the Root Mean Square Error (RMSE), which is widely used for rating prediction in recommender systems, to biased Root Mean Square Error (bRMSE). Given a predicted rating $\hat{r}_{ui}$ and a ground-truth rating r_{ui} from a user u for an item i, the RMSE is calculated as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{U\times I}(\hat{r}_{ui} - r_{ui})^2}. \tag{16}$$

Same as $loss_2$ in Section III-E, RRRE defines bRMSE that only takes ratings of benign reviews into consideration:

$$bRMSE = \sqrt{\frac{1}{N_{benign}}\sum_{U\times I} l_{ui}(\hat{r}_{ui} - r_{ui})^2}\ , \tag{17}$$

where $N_{c=benign}$ means the number of benign reviews. As for evaluating the result of rating prediction, a lower b-RMSE score indicates a better performance.

The reliability score is used to select the proper reviews as the explanations of the recommended result. So we would rather formulate this task as a ranking problem than a classification problem. To evaluate the performance, the commonly-used average precision (AP) and area under the ROC curve (AUC) are employed as the evaluation metrics. The AUC score, which always measures the performance of supervised classification tasks, is a standard measure when data is imbalanced, as is our case.

[2]http://$jmcauley.ucsd.edu/data/amazon/$

For reliability score prediction, the quality at the top of the ranking results is the most important. Therefore, the value of NDCG@k is also employed to measure the performance of ranking the reliable reviews, and it is formally defined as follows:

$$NDCG@\mathtt{k}v = \frac{DCG@\mathtt{k}}{IDCG@\mathtt{k}}, \tag{18}$$

$$DCG@\mathtt{k} = \sum_{i=1}^{\mathtt{k}} \frac{2^{l_i} - 1}{log_2(i+1)}, \tag{19}$$

where l_i captures the true label of the review at rank i (1: benign review, 0: fake review) and $IDCG@\mathtt{k}$ is the DCG for ideal ranking where all l_i's are 1. Following [20], k varies from 100 to 1000.

B. Baselines

To prove the performance of RRRE, the following baseline algorithms are selected from the outstanding methods described in Section II. The rating score prediction and the reliability score prediction are introduced respectively:

1) Rating Prediction:

- **PMF** [41]: This is a traditional matrix factorization method, and the model parameters are learned by stochastic gradient descent (SGD).
- **DeepCoNN** [42]: This is the first joint deep learning framework that model users and items from textual reviews for recommendation. The authors have shown that it can achieve significant improvement compared with other strong topic modeling based methods.
- **NARRE** [9]: This is an explainable recommendation method considering the usefulness of reviews. We implemented it based on the authors' public code.
- **DER** [10]: This is a state-of-the-art explainable recommendation method, which considers time feature and has been verified to outperform many promising algorithms including GRU4Rec, Time-LSTM, Time-LSTM++ on Amazon and Yelp datasets.
- **RRRE$^-$**: The RRRE uses Equation (13) instead of Equation (14) for training. The performance of $RRRE^-$ will show the importance of reliable reviews to rating prediction.

2) Reliability Score Prediction:

- **ICWSM13** [43]: This algorithm adds behavioral features of users to improve the performance.
- **SpEagle+** [20]: A supervised extension of the belief propagation-based algorithms SpEagle [20] and FraudEagle [16].

TABLE III
PERFORMANCE OF RATING PREDICTION: THE TABLE SHOWS THE BRMSE VALUES OF ALL ALGORITHMS IN RATING PREDICTION

	RRRE	PMF	DeepCoNN	NARRE	DER	$RRRE^-$
YelpChi	**0.965**	1.052	0.994	1.002	1.112	1.041
YelpNYC	**0.989**	1.081	0.992	1.030	1.048	1.058
YelpZip	**0.983**	1.101	1.092	1.073	1.087	1.062
Musics	**1.054**	1.194	1.143	1.156	1.170	1.179
CDs	**0.977**	1.081	0.998	1.060	1.088	1.098

TABLE IV
PERFORMANCE OF RELIABILITY SCORE PREDICTION: AUC VALUES AND AVERAGE PRECISION OF ALL ALGORITHMS.

	AUC					Average Precision				
	Musics	CDs	YelpChi	YelpNYC	YelpZip	Musics	CDs	YelpChi	YelpNYC	YelpZip
ICWSM13	0.734	0.722	0.713	0.654	0.632	0.857	0.869	0.856	0.843	0.895
SpEagle+	0.759	0.763	**0.795**	0.783	0.804	0.416	0.405	0.397	0.348	0.425
REV2	0.798	0.803	0.625	0.648	0.634	0.801	0.819	0.532	0.503	0.612
RRRE	**0.911**	**0.924**	0.789	**0.791**	**0.806**	**0.965**	**0.977**	**0.956**	**0.929**	**0.934**

- **REV2** [19]: Three metrics are proposed to rank users, items, and ratings respectively. REV2 iteratively computes these metrics and uses Bayesian approaches to address cold start problems.

RRRE uses the textual content of a review, user ID, and item ID as input features. Besides, rating time distribution, metadata of a review (e.g., length of text, specific symbol), and user attributes (e.g., gender) are given to evaluate baseline methods.

C. Performance of Rating Prediction

In this experiment, the task is to give a predicted rating to each review. In real scenarios, it is meaningless to predict fake ratings and recommend items to the users who write fake reviews. For this reason, this paper compares RRRE with the baseline algorithms in terms of the proposed bRMSE score, which only evaluates the performance of rating prediction on benign reviews. For each data set, 70% of instances are used to train the model and 30% for testing. Both training data and testing data are the subsets of $U \times I$. Each user or item has at least one review. In TABLE III, the results are the mean values of five experiments.

The results of bRMSE show that the performance of RRRE is better than the baseline methods. The reason is that, during the training process, RRRE only learns from benign reviews by using bRMSE as the optimization function while the other algorithms are confused by fake reviews. RRRE models more adaptive and reasonable profiling of users (items) by distinguishing the reliability of reviews, and eventually improves the performance of rating prediction. RRRE performs better on YelpChi than that on other datasets. We think there are two reasons. Firstly, YelpZip and YelpNYC are large-scale datasets and bring more noise to the model. Secondly, the low degree of items in Amazon datasets reduces the information of the item profiles.

We train RRRE with Equation (14) and $RRRE^-$ with Equation (13). The results in TABLE III show that RRRE performs better than $RRRE^-$ and prove that considering the reliability score improves the performance of recommendation.

As a state-of-the-art method, DER should have achieved a better performance than the reported results in TABLE III. The most possible reason is that each user in each dataset has less than three reviews on average, and each item in Amazon datasets has less than three reviews on average (as shown in TABLE II). The low degree of users and items makes this dynamic model hard to bring into full play.

TABLE V
NDCG@k OF COMPARED METHODS ON YELPCHI

k	ICWSM13	SpEagle+	REV2	RRRE
100	0.567	0.975	0.432	**0.989**
200	0.551	0.962	0.425	**0.986**
300	0.546	0.951	0.419	**0.986**
400	0.541	0.938	0.406	**0.982**
500	0.532	0.924	0.395	**0.979**
600	0.535	0.905	0.386	**0.972**
700	0.525	0.889	0.389	**0.967**
800	0.511	0.865	0.376	**0.959**
900	0.486	0.849	0.374	**0.951**
1000	0.459	0.835	0.364	**0.940**

TABLE VI
NDCG@k OF COMPARED METHODS ON CDS

k	ICWSM13	SpEagle+	REV2	RRRE
100	0.488	0.921	0.554	**0.998**
200	0.465	0.906	0.545	**0.991**
300	0.470	0.885	0.542	**0.985**
400	0.454	0.884	0.536	**0.974**
500	0.438	0.875	0.532	**0.971**
600	0.435	0.860	0.524	**0.966**
700	0.424	0.858	0.515	**0.956**
800	0.417	0.855	0.516	**0.950**
900	0.401	0.824	0.494	**0.936**
1000	0.392	0.801	0.482	**0.927**

D. Performance of Reliability Score Prediction

In this experiment, both RRRE and baseline methods calculate a probability score for each review and evaluate the result with a boolean label. TABLE IV records the AP and AUC

performance of RRRE and baselines across datasets. TABLE V and TABLE VI record $NDCG@k$ results.

The results in TABLE IV show that RRRE performs the best in five out of six cases and second-best in the remaining one. RRRE achieves the best performance in both TABLE V and TABLE VI. We can see that there is no consistently well-performing existing algorithm. Data is a very important factor that influences the performance of algorithms. REV2 is a very advanced graph-based algorithm, but the datasets only have a part of reviews for items and users. The low degree of users and items leads to a sparse network and the worse performance. SpEagle+ utilizes graph as well as metadata (text, timestamp, rating) and achieves a better performance. RRRE beats other methods because it makes full use of the textual content, which has rich semantic information, to profile the users and items. Every algorithm performs better on Amazon datasets, which are more balanced than Yelp datasets.

E. Evaluation of Hyper-parameter

1) Dimensions of Review Embedding: In this section, we investigate how the embedding size of a review, hyper-parameter k, influences our model's performance. Due to the space limitation, unless specified, we only report the results on the YelpChi dataset. We observe the performance changes of the training process by tuning the embedding size k in the range of 8, 16, 32, 64, 128. The results of rating score prediction and reliability score prediction are recorded in two subfigures respectively.

From the results presented in Fig. 2, we can find that RRRE achieves better performances when the embedding size is increasing (i.e., k = 64) in both figures, while larger k doesn't help to further improve the results. The reason for this observation can be that: in our dataset, the semantic information of textual content is hard to be captured by a smaller embedding size. So $k = 64$ performs better than $k = 8$ obviously. But the similar lines of $k = 64$ and $k = 128$ in the subfigures indicate that the benefits from increasing embedding size are limited. Using redundant dimensions will increase the model complexity and overfit the training set, which may degrade our model's generalization capability on the testing set. Thus, the most suitable embedding size is set to $k = 64$ in our model.

2) Number of reviews in input layer: In this section, we investigate how the size of the input layer, hyper-parameter s, influences the performance of RRRE. A larger s means more information for preference extraction, but it also costs more computing resources. Besides, if $|W^u| > s$ ($|W^i| > s$), the larger s may cause over-fitting problem. If $|W^u| < s$ ($|W^i| < s$), the larger s brings more zero padding and causes sparsity problem. Without expert knowledge, a suitable parameter will be found through experiments.

It's a common sense that the number of an item's reviews is much larger than a user's. In the YelpChi dataset, 158 is the maximum value of $|W^u|$ and the median value is 3. As to $|W^i|$, 473 is the maximum number and the median value is 72. Based on the statistical analysis, RRRE sets s_u for

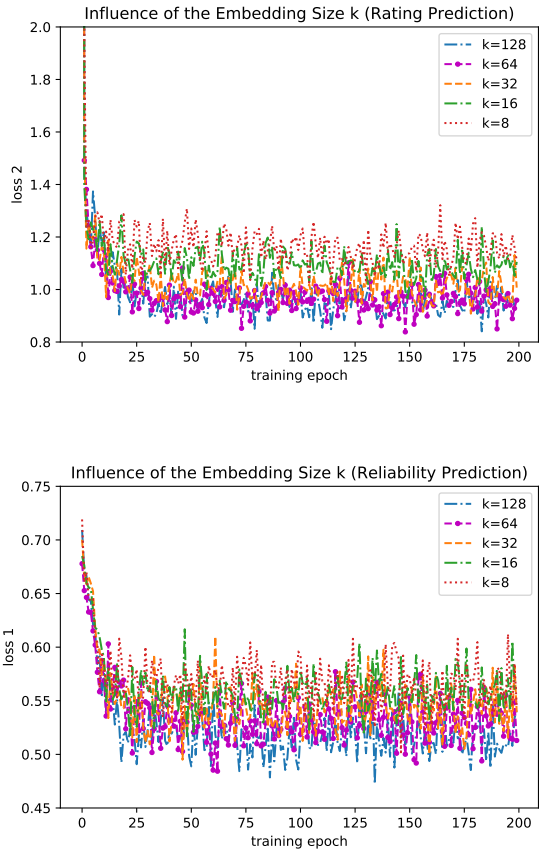

Fig. 2. The Influence of the Embedding Size k for Training Process

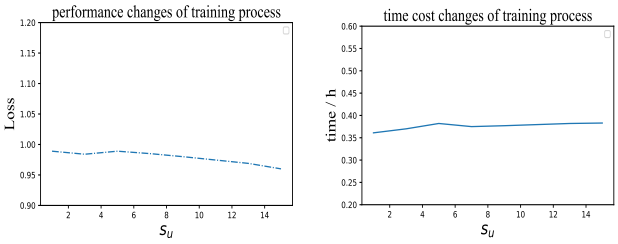

Fig. 3. Influence of the Input Size s_u for Training Process

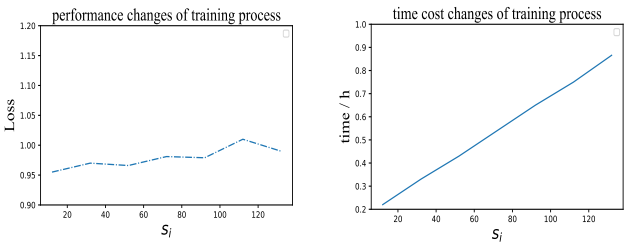

Fig. 4. The Influence of the Input Size s_i for Training Process

TABLE VII
CASE STUDY: RECOMMENDATION RESULT

user ID	item	textual Content	Predicted Rating (Real Rating)	Predicted Reliability (Real Reliability)
zCvaSXHpGox	Beer Bar	friendly service, impressive beer selection! the upstairs (somewhat hidden) bar is the best for weekends.	4.263 (5)	0.763 (1)
zCvaSXHpGox	Hot Doug	everyone knows how amazing it is Doug is incredibly friendly. The staff is incredibly friendly	4.154 (5)	0.901 (1)
zCvaSXHpGox	Light Bar	My experience wasn't all that great. I felt pressured to try one of their microbrews when all I wanted was a soda	3.890 (4)	0.876 (1)

TABLE VIII
CASE STUDY: RELIABLE EXPLANATION

item	user ID	textual Content	Predicted Rating (Real Rating)	Predicted Reliability (Real Reliability)
Hot Doug	YLCb4dS3vw	Walking into Hot Doug's, I had pretty high expectations from all the hype. And this place definitely lived up to the hype To me, it was the sausage that made it great	4.476 (5)	0.892 (1)
Hot Doug	Y-L5qP02Q7	I don't think I ever left this place dissapointed. Goddammit,Hot Dougs. I'm addicted to your corn dogs now.	4.531 (5)	0.405 (0)

$UserNet$ and s_i for $ItemNet$ respectively. In this experiment, the median is the reference for our experimental parameter setting. We observe the performance changes of the training process by setting the s_i as 72 and tuning the s_u in the range of 1, 3, 5, 7, 9, 11, 13. Similarly, we also set the s_u as 11 and tune the s_i in the range of 12, 32, 52, 72, 92, 112, 132. In order to get a reliable conclusion, other experimental parameters are fixed ($k = 64$, $batch\_size = 500$, $training\_epoch = 1000$ and the same GPU).

From the results presented in Fig. 3 and Fig. 4, we can find that the two groups of experiments show different conclusions. With the increase of s_u, the performance of the model improves slowly and the time cost changes a little. Our explanation is that the average $|W^u|$ is too small for the model to get more information about the user preferences, and zero padding won't take too much time. With the increase of s_i, the performance of the model is gradually rising and the time cost is growing linearly. Our explanation is that the average $|W^i|$ is much larger than s_i. Over-fitting and heavy sampling decrease performance and increase the time cost. Taking performance and time cost into account, $s_u = 13$ and $s_i = 12$ are the good settings in current experiments.

F. Case Study: Recommendation with Reliable Explanation

To evaluate the generation process of recommendation results and explanations described in Section III-B. TABLE VII and TABLE VIII show a case study that attempts to recommend an item to a user $'zCvaSXHpGox'$ with corresponding explanations respectively. In order to save space, some long textual contents are partly selected.

Recommendation Result: We calculate the rating score and the reliability score between the user $'zCvaSXHpGox'$ and all items. Then the rating scores are sorted, and the details of the top three candidates are shown in TABLE VII. The item $'HotDoug'$ has the highest reliability score 0.901 and is selected as the recommendation result.

Reliable Explanation: For the item $'HotDoug'$, we calculate the rating score and the reliability score of all the reviews that are written to $'HotDoug'$. Then the rating scores are sorted, and the details of the top two candidates are shown in TABLE VIII. The reviews written from $'YLCb4dS3vw'$ will be shown to the user $'zCvaSXHpGox'$ as the explanation along with the recommended item $'HotDoug'$. Although the second review has a higher rating score, it will be filtered because of its low reliability.

V. CONCLUSION

In real application scenarios, a lot of fake reviews unjustly promote or demote certain products or items. It decreases the performance of the explainable recommender system in both recommendation and explanation. This paper proposes a model named Reliable Recommendation with Review-level Explanations, which promotes rating prediction by detecting fake reviews. We evaluated our method on three real-world datasets with labeled reviews. Our results show that RRRE is superior to predict rating when the fake reviews existed. By calculating the reliability scores of reviews, RRRE can provide customers with reliable recommendation results and corresponding explanations.

This is a first step to combine the two subtasks into a uniform model, and there is much room for further improvements. In the future, we will focus on improving the work style of the model. At first, if the size of datasets goes large, RRRE may need very heavy sampling. This will severely degrade the performance of the model. Second, we will improve the design of our model to facilitate semi-supervised learning so that it can easily adapt to new users and items without any review in our future work.

VI. Acknowledgments

This work is supported by National Key R&D Program of China (No.2017YFB1401300, 2017YFB1401302), National Natural Science Foundation of China (No.61672419, 61672418, 61872287, 61877050 and 61937001), MoE-CMCC "Artificial Intelligence" Project No.MCM20190701, Innovative Research Group of the National Natural Science Foundation of China (No.61721002), Innovation Research Team of Ministry of Education (IRT_17R86), Project of China Knowledge Centre for Engineering Science and Technology, the consulting research project of Chinese academy of engineering "The Online and Offline Mixed Educational Service System for 'The Belt and Road' Training in MOOC China". This work is also supported by Australian Research Council (Grant No.DP190101985, DP170103954).

References

[1] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 426–434.

[2] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, no. 8, pp. 30–37, 2009.

[3] T. Chen, H. Yin, G. Ye, Z. Huang, Y. Wang, and M. Wang, "Try this instead: Personalized and interpretable substitute recommendation," *arXiv preprint arXiv:2005.09344*, 2020.

[4] S. Zhang, H. Yin, T. Chen, Q. V. N. Hung, Z. Huang, and L. Cui, "Gcn-based user representation learning for unifying robust recommendation and fraudster detection," *arXiv preprint arXiv:2005.10150*, 2020.

[5] X. He, T. Chen, M.-Y. Kan, and X. Chen, "Trirank: Review-aware explainable recommendation by modeling aspects," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. ACM, 2015, pp. 1661–1670.

[6] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma, "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 2014, pp. 83–92.

[7] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, and Y. Liu, "Explicit factor models for explainable recommendation based on phrase-level sentiment analysis," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 2014, pp. 83–92.

[8] X. Chen, Z. Qin, Y. Zhang, and T. Xu, "Learning to rank features for recommendation over multiple categories," in *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*. ACM, 2016, pp. 305–314.

[9] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations," in *Proceedings of the 2018 World Wide Web Conference*. International World Wide Web Conferences Steering Committee, 2018, pp. 1583–1592.

[10] X. Chen, Y. Zhang, and Z. Qin, "Dynamic explainable recommendation based on neural attentive models." AAAI, 2019.

[11] M. Jiang, P. Cui, and C. Faloutsos, "Suspicious behavior detection: Current trends and future directions," *IEEE Intelligent Systems*, vol. 31, no. 1, pp. 31–39, 2016.

[12] S. Kumar and N. Shah, "False information on web and social media: A survey," *arXiv preprint arXiv:1804.08559*, 2018.

[13] Y. Li, Z. Zhang, Y. Peng, H. Yin, and Q. Xu, "Matching user accounts based on user generated content across social networks," *Future Generation Computer Systems*, vol. 83, pp. 104–115, 2018.

[14] Y. Xu, Y. Yang, J. Han, E. Wang, F. Zhuang, and H. Xiong, "Exploiting the sentimental bias between ratings and reviews for enhancing recommendation," in *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2018, pp. 1356–1361.

[15] G. Ramos, L. Boratto, and C. Caleiro, "On the negative impact of social influence in recommender systems: A study of bribery in collaborative hybrid algorithms," *Information Processing & Management*, vol. 57, no. 2, p. 102058, 2020.

[16] L. Akoglu, R. Chandy, and C. Faloutsos, "Opinion spam detection in online reviews by network effects," in *Seventh international AAAI conference on weblogs and social media*, 2013.

[17] Y. Zhang and X. Chen, "Explainable recommendation: A survey and new perspectives," *arXiv preprint arXiv:1804.11192*, 2018.

[18] N. Wang, H. Wang, Y. Jia, and Y. Yin, "Explainable recommendation via multi-task learning in opinionated text data," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 2018, pp. 165–174.

[19] S. Kumar, B. Hooi, D. Makhija, M. Kumar, C. Faloutsos, and V. Subrahmanian, "Rev2: Fraudulent user prediction in rating platforms," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 2018, pp. 333–341.

[20] S. Rayana and L. Akoglu, "Collective opinion spam detection: Bridging review networks and metadata," in *Proceeding of the 21st ACM SIGKDD international conference on Knowledge discovery and data mining, KDD15*, 2015.

[21] A. Mishra and A. Bhattacharya, "Finding the bias and prestige of nodes in networks based on trust scores," in *Proceedings of the 20th international conference on World wide web*. ACM, 2011, pp. 567–576.

[22] G. Wang, S. Xie, B. Liu, and P. S. Yu, "Identify online store review spammers via social review graph," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 4, p. 61, 2012.

[23] Z. Wu, C. C. Aggarwal, and J. Sun, "The troll-trust model for ranking in signed networks," in *Proceedings of the Ninth ACM international conference on Web Search and Data Mining*. ACM, 2016, pp. 447–456.

[24] J. Meng, C. Peng, A. Beutel, C. Faloutsos, and S. Yang, "Catchsync: Catching synchronized behavior in large directed graphs," in *Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, 2014.

[25] A. Fayazi, K. Lee, J. Caverlee, and A. Squicciarini, "Uncovering crowdsourced manipulation of online reviews," in *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. ACM, 2015, pp. 233–242.

[26] V. Sandulescu and M. Ester, "Detecting singleton review spammers using semantic similarity," in *Proceedings of the 24th international conference on World Wide Web*. ACM, 2015, pp. 971–976.

[27] H. Sun, A. Morales, and X. Yan, "Synthetic review spamming and defense," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 1088–1096.

[28] Y. Ren and D. Ji, "Neural networks for deceptive opinion spam detection," *Information Sciences*, vol. 385, pp. 213–224, 2017.

[29] A. Mukherjee, A. Kumar, B. Liu, J. Wang, M. Hsu, M. Castellanos, and R. Ghosh, "Spotting opinion spammers using behavioral footprints," pp. 632–640, 2013.

[30] S. Xie, G. Wang, S. Lin, and P. S. Yu, "Review spam detection via temporal pattern discovery," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 823–831.

[31] B. Pal and M. Jenamani, "Trust inference using implicit influence and projected user network for item recommendation," *Journal of Intelligent Information Systems*, vol. 52, no. 2, pp. 425–450, 2019.

[32] P. Massa and B. Bhattacharjee, "Using trust in recommender systems: an experimental analysis," in *International conference on trust management*. Springer, 2004, pp. 221–235.

[33] M. Jamali and M. Ester, "Trustwalker: a random walk model for combining trust-based and item-based recommendation," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 397–406.

[34] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 2009, pp. 203–210.

[35] B. Yang, Y. Lei, J. Liu, and W. Li, "Social collaborative filtering by trust," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 8, pp. 1633–1647, 2016.

[36] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel recommendation model regularized with user trust and item ratings," *ieee transactions on knowledge and data engineering*, vol. 28, no. 7, pp. 1607–1620, 2016.

[37] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *Journal of machine learning research*, vol. 12, no. Aug, pp. 2493–2537, 2011.
[38] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.
[39] H. Yin, Q. Wang, K. Zheng, Z. Li, J. Yang, and X. Zhou, "Social influence-based group representation learning for group recommendation," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 2019, pp. 566–577.
[40] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*. International World Wide Web Conferences Steering Committee, 2017, pp. 173–182.
[41] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Advances in neural information processing systems*, 2008, pp. 1257–1264.
[42] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*. ACM, 2017, pp. 425–434.
[43] A. Mukherjee, V. V. Venkataraman, B. Liu, and N. S. Glance, "What yelp fake review filter might be doing," pp. 409–418, 2013.